A machine learning approach for result caching in web search engines

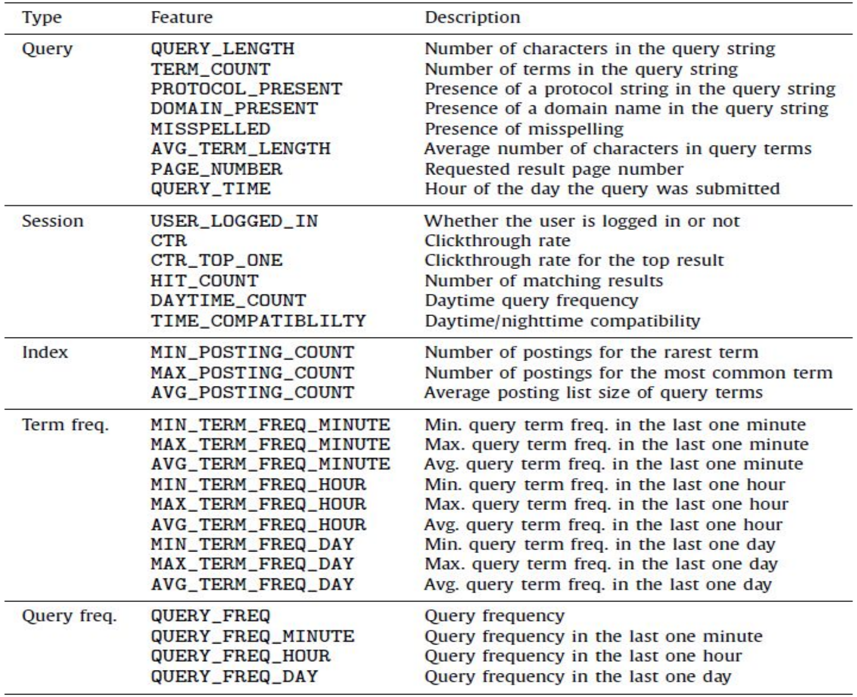
Tayfun Kucukyilmaz, B. Barla Cambazoglu, Cevdet Aykanat, Ricardo Baeza-Yates

Presented By: Rushikesh Padia

Overview

- Proposes machine learning based approaches for Static, Dynamic, Static-Dynamic Cache
- Proposes unifying framework which uses extensive set of features.
- Applies variety of models to evaluate impact of Hit Rate
- Static Cache modeling: Offline cache allocation problem
- Dynamic Cache modeling: Online Eviction Problem
- Uses classical ML models

Features



Type	Feature	Description
Query	QUERY_LENGTH	Number of characters in the query string
	TERM_COUNT	Number of terms in the query string
	PROTOCOL_PRESENT	Presence of a protocol string in the query string
	DOMAIN_PRESENT	Presence of a domain name in the query string
	MISSPELLED	Presence of misspelling
	AVG_TERM_LENGTH	Average number of characters in query terms
	PAGE_NUMBER	Requested result page number
	QUERY_TIME	Hour of the day the query was submitted
Session	USER_LOGGED_IN	Whether the user is logged in or not
	CTR	Clickthrough rate
	CTR_TOP_ONE	Clickthrough rate for the top result
	HIT_COUNT	Number of matching results
	DAYTIME_COUNT	Daytime query frequency
	TIME_COMPATIBLILTY	Daytime/nighttime compatibility
Index	MIN_POSTING_COUNT	Number of postings for the rarest term
	MAX_POSTING_COUNT	Number of postings for the most common term
	AVG_POSTING_COUNT	Average posting list size of query terms
Term freq.	MIN_TERM_FREQ_MINUTE	Min. query term freq. in the last one minute
	MAX_TERM_FREQ_MINUTE	Max. query term freq. in the last one minute
	AVG_TERM_FREQ_MINUTE	Avg. query term freq. in the last one minute
	MIN_TERM_FREQ_HOUR	Min. query term freq. in the last one hour
	MAX_TERM_FREQ_HOUR	Max. query term freq. in the last one hour
	AVG_TERM_FREQ_HOUR	Avg. query term freq. in the last one hour
	MIN_TERM_FREQ_DAY	Min. query term freq. in the last one day
	MAX_TERM_FREQ_DAY	Max. query term freq. in the last one day
	AVG_TERM_FREQ_DAY	Avg. query term freq. in the last one day
Query freq.	QUERY_FREQ	Query frequency
	QUERY_FREQ_MINUTE	Query frequency in the last one minute
	QUERY_FREQ_HOUR	Query frequency in the last one hour
	QUERY_FREQ_DAY	Query frequency in the last one day

Static Result Caching

- Baseline | Offline-LRU
- Baseline | MFU
- Baseline | QDEV (Query stability)
- Oracle | Theoretical Oracle (TO): Selects Most frequently used in test set
- Oracle | Practical Oracle (PO): Selects Most frequently used in test set which also appears in training set

Static Result Caching

Machine learned static caching (MLSC)

- Uses regression model to predict IAT-NEXT (next occurrence time)
- lower IAT_NEXT, earlier it will appear; higher the value, later it will appear.
- MLSC & Off-LRU are similar. Off-LRU uses training set frequencies, MLSC uses predicted frequencies of past queries
- Based on assumption, query carries characteristics markers which can be extracted by ML algorithms

Dynamic Result Caching

- Baseline | LRU
- Oracle | Belady: Clairvoyant algorithm or optimal algorithm
- Machine learned Dynamic Caching (MLDC): predicts IAT_NEXT for the queries
- Uses 2 classifier approach
  - Singleton Classifier: Predicts [0, 1] for singleton queries. Singleton queries are queries that appear only once.
  - Non-singleton Classifier: Fits regression model, where the class labels is IAT_NEXT

Dynamic Result Caching

- Approach 1: Use Singleton Classifier for admission, Non-singleton for eviction
- Approach 2: Uses linear combination of both classifier to take eviction decision
- Segmentation: Uses segmentation approach to prevent permanent pollution of cache

Static-Dynamic Cache

- Baseline | Static-Dynamic Cache (SDC)
- Oracle | SDC - dynamic oracle | SDC-Belady
- Oracle | SDC - static oracle | SDC-PO
- Proposed | MLSC+LRU / Off-LRU+MLDC
- Proposed | MLSDC

Dataset

- Yahoo 10 days data
- AOL query logs https://jeffhuang.com/search_query_logs/
- ML Algos - MLP, pace regression, SVM, KNN, logistic regression, Gradient Boosted Decision Tree
- Best performance by GBDT

Reference

[1] T. Kucukyilmaz, B. B. Cambazoglu, C. Aykanat, and R. Baeza-Yates, â€œA machine learning approach for result caching in web search engines,â€□ Information Processing & Management, vol. 53, no. 4, pp. 834â€“850, 2017, doi: https://doi.org/10.1016/j.ipm.2017.02.006.